



CICE - La Escuela Profesional de nuevas Tecnologías  
Master de Programación en Python

# **INDEXED WORLD**

**[www.mundoindexado.com](http://www.mundoindexado.com)**

Integrantes:

Isabel Repetto García-Plata

Verónica Salguero Herrera

Madrid, Junio de 2021.

## Contenido

1. OBJETIVO PRINCIPAL .....	2
2. OBJETIVOS ESPECÍFICOS.....	2
3. ALCANCE DEL PROYECTO .....	3
4. LÍMITES DEL PROYECTO.....	4
5. ASPECTOS MAS IMPORTANTES DEL PROYECTO.....	5
5.1 SERVICIO .....	5
5.2 RESTRICCIONES .....	5
5.3 FUNCIONALIDAD .....	5
5.4 RENDIMIENTO .....	8
5.5 HERRAMIENTAS Y TÉCNICAS .....	8
6. ANEXO .....	12
7. Bibliografía .....	15



# 1. OBJETIVO PRINCIPAL

Nuestro objetivo principal es crear una aplicación web que permita al usuario hacer búsquedas de cualquier producto electrónico deseado para obtener solo resultados con descuento aplicado, después de haber realizado una búsqueda comparativa entre distintas webs.

## 2. OBJETIVOS ESPECÍFICOS

- ❖ Introducir los conceptos básicos de programación en python.
- ❖ Conocer algoritmos fundamentales de las ciencias de la computación.
- ❖ Desarrollar la capacidad para analizar, diseñar e implementar soluciones computacionales de baja y media complejidad.
- ❖ Mejorar las habilidades creativas, comunicativas y colaborativas, ya que, al trabajar en equipo, estamos en proceso constante de adaptación.
- ❖ Poder hacer estudios de mercado en el ámbito seleccionado (productos electrónicos)
- ❖ Conocer los principales componentes software para trabajar con python, y familiarizarse con la interfaz, en este caso, Django.
- ❖ Utilizar con seguridad y eficacia sistemas operativos de uso común, como son MacOS y Windows.
- ❖ Ser más rápido y eficiente que ningún otro buscador hasta el momento.
- ❖ Crear aplicaciones sencillas de software estructurado utilizando un Entorno de Desarrollo Integrado orientadas a la producción de una página web interactiva.
- ❖ Estudio de la estructura interna de webs como Ebay, PC Componentes, Efecto 2000, MediaMark, FNAC, entre otras, para poder hacer uso de su información como si accediésemos a una base de datos.

### 3. ALCANCE DEL PROYECTO

- ❖ Debe poder desplegarse en distintas plataformas y sistemas operativos sin necesidad de una adaptación previa, en concreto debe ser capaz de ejecutarse al menos en los siguientes sistemas:
  -  Windows
  -  MacOS
- ❖ Debe permitir la comunicación efectiva entre equipos físicamente remotos, dándole la sensación al usuario de estar ejecutando una aplicación local. Para el usuario operador no debe ser necesaria la realización de ninguna tarea adicional a la que realizaría si la aplicación corriera en local.
- ❖ El servicio debe ser fiable e inmune a los problemas de red.
- ❖ Implementar un mecanismo que gestione la concurrencia de peticiones.
- ❖ No debe implicar riesgos de seguridad para las máquinas que lo ejecuten.
- ❖ Diseño e implementación de mecanismos para garantizar la continua ejecución de las partes del sistema que así lo requiera (servicios y scripts).
- ❖ Manejar y analizar la información procedente de distintas webs sin necesidad de disponer de una base de datos propia.
- ❖ Ofrecer al usuario una interfaz responsive. Intuitiva y adaptable a cualquier dispositivo.
- ❖ Realizar cada parte del proyecto programando web scraping y frontend principalmente.
- ❖ Buscar toda la información necesaria a lo largo del proyecto.
- ❖ Realizar toda la documentación relativa a nuestra aplicación.

## 4. LÍMITES DEL PROYECTO

Implementación TSA por no cumplir requisitos previos, como disponer de un dominio. Por lo que la aplicación, para este proyecto, solo correrá en local.

Implementación en IOS o Android, debido a la complejidad que supone en cuanto al tiempo que disponemos para realizar nuestro proyecto.

Poca experiencia en la creación y desarrollo de proyectos.

Como trabajo en grupo, contamos con el limitante del tiempo libre común disponible, ya que las dos personas que componemos este grupo estamos trabajando, cada una con un horario diferente. No podremos hacer reuniones diarias para el seguimiento del proyecto con lo que la falta de comunicación será uno de nuestros principales hándicaps.

Extracción de datos de <https://www.amazon.es> . Presenta una cantidad considerable de obstáculos a la hora de acceder a sus datos y puesto que tenemos un tiempo limitado, no nos queda otra que prescindir, por el momento, de los productos de esta web.

Además, hemos sido bloqueadas por otras webs como [www.pccomponentes.es](http://www.pccomponentes.es), [www.worten.es](http://www.worten.es), [www.ebay.es](http://www.ebay.es), [www.fnac.es](http://www.fnac.es), etc, tras la detección de la presencia de bots o en alguna otra nos hemos encontrado con cookies que no nos han permitido el acceso a la información.

## **5. ASPECTOS MAS IMPORTANTES DEL PROYECTO**

### **5.1 SERVICIO**

Nuestra web ofrece el servicio de búsqueda de productos electrónicos en diferentes webs y rescate de todos aquellos que tengan aplicado cualquier tipo de descuento. Es decir, si el cliente quiere comprar un producto en concreto, solo tiene que utilizar este servicio para encontrar la mejor oferta sin necesidad de invertir mas de 2 minutos en comparar nada.

### **5.2 RESTRICCIONES**

Al ser tan solo un proyecto de finalización de estudios y no un proyecto real, no hemos adquirido un dominio ni un servidor para hacerlo correr en la web con lo que la mayor restricción real es que el proyecto correrá en local.

Otra de las restricciones que presenta este servicio es que todo producto en oferta que extraigamos de [www.mediamarkt.es](http://www.mediamarkt.es) no dispone de imagen porque la web nos deniega el acceso a ese dato, con lo cual obtendremos el resto de la información con una divertida imagen advirtiéndole que no dispone de la foto del producto.

### **5.3 FUNCIONALIDAD**

El funcionamiento de la web es bastante intuitivo ya que nos hemos encargado de que el formulario de búsqueda quede centrado y señalado en la página principal.


¡Busca tu ganga!

Barra de búsqueda


raton

Search

## ¿Cansado de pagar mas por el mismo producto?



No seas tonto que no se te escapen las gangas



Extraemos las mejores ofertas de las mejores webs de electrónica



¿Estas buscando un nuevo dispositivo Android y no quieres pagar de mas?



¿O quizás seas mas de Apple?


La web solo dispone de dos interfaces. La principal, que es la que ofrece el formulario de búsqueda y otras búsquedas directas a los productos más buscados.

**Indexed World**


[Home](#) [About](#)

CATEGORIAS


### Lo más buscado




Auriculares




Ratones




Teclados



Smartphone



Consolas



Accesorios

También ofrece un enlace, aparte del de la barra de navegación, que da acceso a la otra cara de la web que es el “About”, dónde nos presentamos y hablamos un poco sobre el proyecto en sí.

# Indexed World

[Home](#)

Somos

## Isa y Vero

Estudiantes del master de  
Python en CICE

### Cronología del Proyecto



### ¿Quiénes Somos?

Somos dos estudiantes del master en Python de la escuela de CICE, este es nuestro proyecto final de master

[Pruébalo](#)

Mundo-Indexado  
Calle Falsa 1234  
veronicasalguero.95@gmail.com  
O





## 5.4 RENDIMIENTO

Al tratarse de datos extraídos directamente de la web y no de una base de datos, el tiempo de respuesta es mayor de lo habitual. Las búsquedas precisan una media de 9 segundos para mostrar la respuesta. Debemos tener en cuenta que estamos haciendo “scraping” a varias webs, que supone la extracción del html o estructura de la página, la búsqueda de las etiquetas y/o atributos dentro de esa estructura dónde se encuentran todos y cada uno de los datos que nosotras necesitamos para mostrarlos en nuestra web.

Estos datos son: el precio actual, el precio antes de que se la haya aplicado el descuento, el porcentaje de descuento, la url donde se encuentra el producto y la imagen (excepto en el caso de MediaMarkt que no nos deja acceder a esto último).

Para llevar a cabo todos estos pasos de una manera mas eficiente, hemos usado hilos, que permiten que las búsquedas de cada una de las webs se realicen de forma concurrente, es decir, que empiecen una detrás de la otra, pero sin necesidad de que la anterior haya finalizado.

## 5.5 HERRAMIENTAS Y TÉCNICAS

### 5.5.1 DJANGO

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin preocuparte de nada mas. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Los motivos principales para usar Django son:

- **Es muy rápido:** Si tenéis una startup, tenéis prisa por terminar vuestro proyecto o, simplemente, queréis reducir costes, con Django podéis construir una aplicación muy buena en poco tiempo.
- **Viene bien cargado:** Cualquier cosa que necesitéis realizar, ya estará implementada, sólo hay que adaptarla a vuestras necesidades. Ya sea porque hay módulos de la comunidad, por cualquier paquete Python que encontréis o las propias aplicaciones que Django trae, que son muy útiles.
- **Es bastante seguro:** Podemos estar tranquilos con Django, ya que implementa por defecto algunas medidas de seguridad, las más clásicas, para que no haya SQL Injection, no haya Cross site request forgery (CSRF) o no haya Clickjacking por JavaScript. Django se encarga de manejar todo esto de una manera realmente sencilla.
- **Es muy escalable:** Podemos pasar desde muy poco a una aplicación enorme perfectamente, una aplicación que sea modular, que funcione rápido y sea estable.
- **Es increíblemente versátil:** Es cierto que en un principio Django comienza siendo un Framework para almacenar noticias por sitios de prensa, blogs y este estilo de webs, pero con el tiempo ha ganado tanta popularidad que se puede usar para el propósito que queráis.

Otras bondades de Django que no se destacan en la web son:

Su ORM, **su interfaz para acceso a la base de datos**, ya que hacer consultas con ella es una maravilla, es una herramienta muy buena.

Trae de serie un panel de administración, con el cual podemos dejar a personas sin ningún tipo de conocimiento técnico manejando datos importantes de una forma muy cómoda.

### 5.5.2 WEB SCRAPING

Web scraping es un término que engloba la utilización de distintos métodos para la recolección de datos a través de Internet. Generalmente, esto se hace con software que simula ser un humano navegando para obtener específicos bits de información desde distintas páginas web.

Aquellos que usan web scraping suelen hacerlo para recolectar ciertos datos que venderán a otros posteriormente, o para propósitos promocionales en alguna web.

Existen otras nomenclaturas para esta técnica de extracción de datos de Internet, como son: Web data extraction, Screen scraping o Web harvesting.

### 5.5.3 PLANTILLAS

Para la estructuración externa y apariencia de nuestra web hemos usado plantillas gratuitas a partir de las cuales hemos ido adaptando a nuestro gusto. Cuando descargamos la plantilla que queramos usar, suele descargarse una carpeta comprimida que contiene, no solo uno o varios ficheros .html, sino que además incluyen las hojas de estilo (.css) y, como en nuestro caso, también algunos scriptst .js (JavaScript) que puede hacer tu página mucho más atractiva sin necesidad de saber ninguno de los tres lenguajes.

Para que todos estos ficheros funcionasen en nuestro proyecto hemos tenido que ubicarlos en directorios específicos ya que Django trabaja con un directorio “static” que es dónde espera encontrar todos estos archivos. (Ver ANEXO).

#### 5.5.4 PRERREQUISITOS / INSTALACIÓN DE PAQUETES

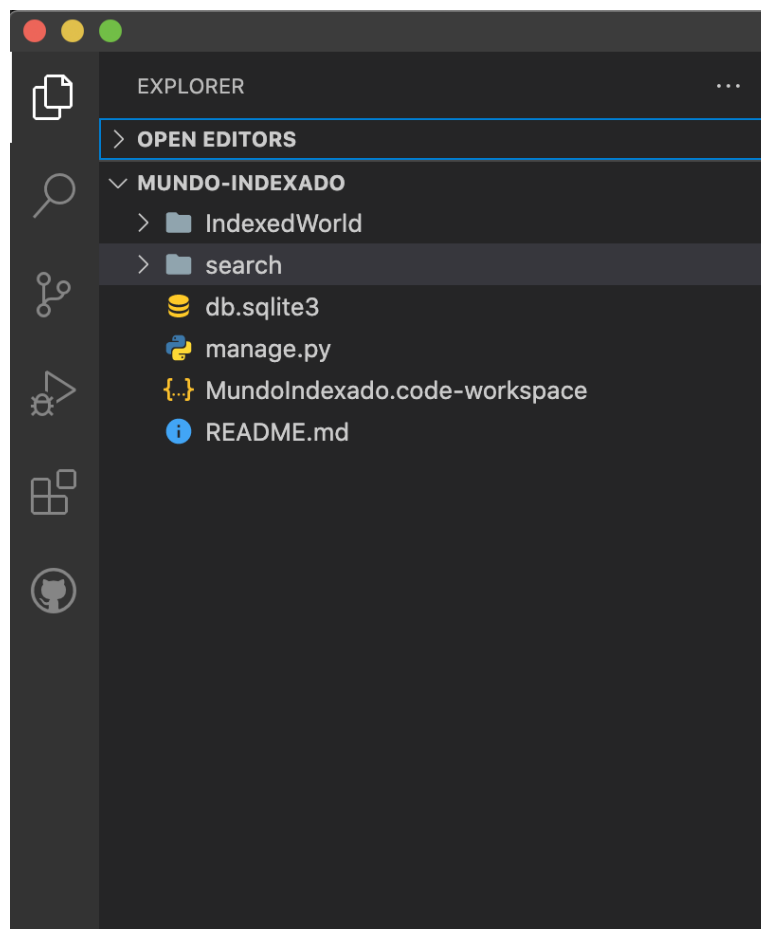
- ❖ **Python** versión 3.8.10.
- ❖ **Visual Studio Code.** Va mucho mas allá que un editor de texto cualquiera. Te permite trabajar en colaboración remota usando la extensión gratuita llamada LiveShare. Su sistema de sugerencias para completar líneas de código, y de corrección rápida de errores comunes, te facilita el tener un código libre de errores gramaticales. También puedes usar su debugger para revisar el funcionamiento de tu código línea a línea y detectar errores rápidamente. Te permite, además, usar Jupiter notebooks con lo que puedes ejecutar tu código paso por paso, visualizarlo e interactuar con tus datos, variables, gráficos...
- ❖ **Django.** Es el framework que se encargará de poner en contacto todos nuestros ficheros. La unificación y/o comunicación entre *back* y *front*.
- ❖ **BeautifulSoup (bs4)** es una biblioteca de Python que sirve para extraer contenido de ficheros HTML y XML.
- ❖ **requests** es una librería que nos permitirá hacer peticiones HTTP entre otras cosas. Hay una librería nativa de Python llamada **urllib2**, que nos provee de múltiples métodos para obtener todo tipo de información de una URL, además de utilidades para normalizar nuestros datos obtenidos, esta librería es sin duda muy útil, pero para un fin más específico que sólo consumir URLs al estilo de cURL, pero también es verdad que una librería de terceros, como es **requests**, nos va a facilitar mucho la tarea.
- ❖ El módulo **concurrent.futures** provee una interfaz de alto nivel para ejecutar invocables de forma asincrónica. La ejecución asincrónica se puede realizar mediante hilos, usando ThreadPoolExecutor, o procesos independientes, mediante ProcessPoolExecutor. Ambos implementan la misma interfaz, que se encuentra definida por la clase abstracta Executor.

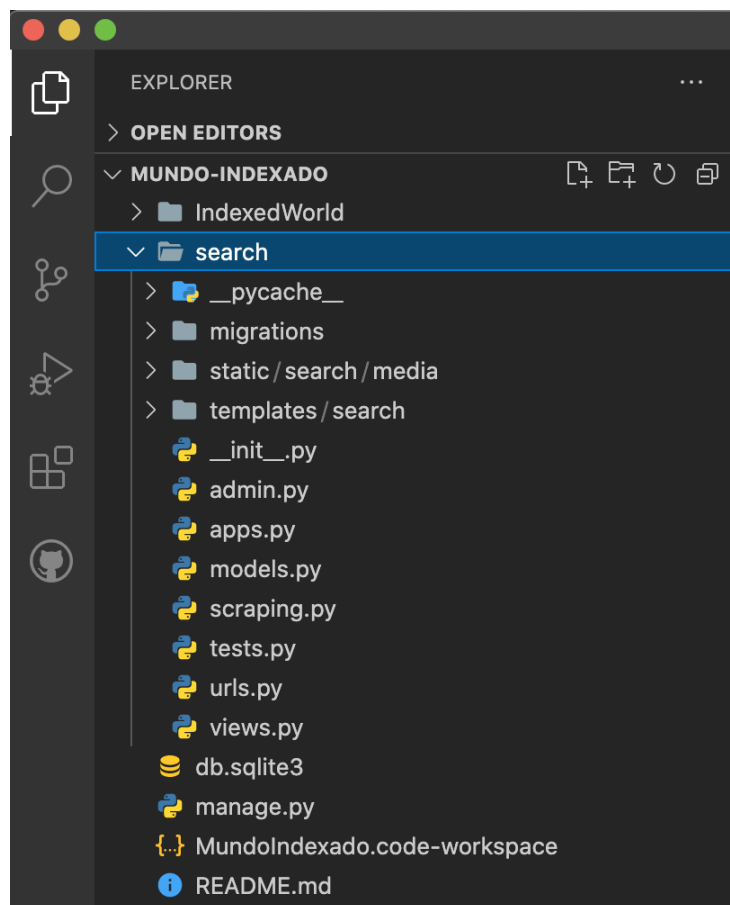
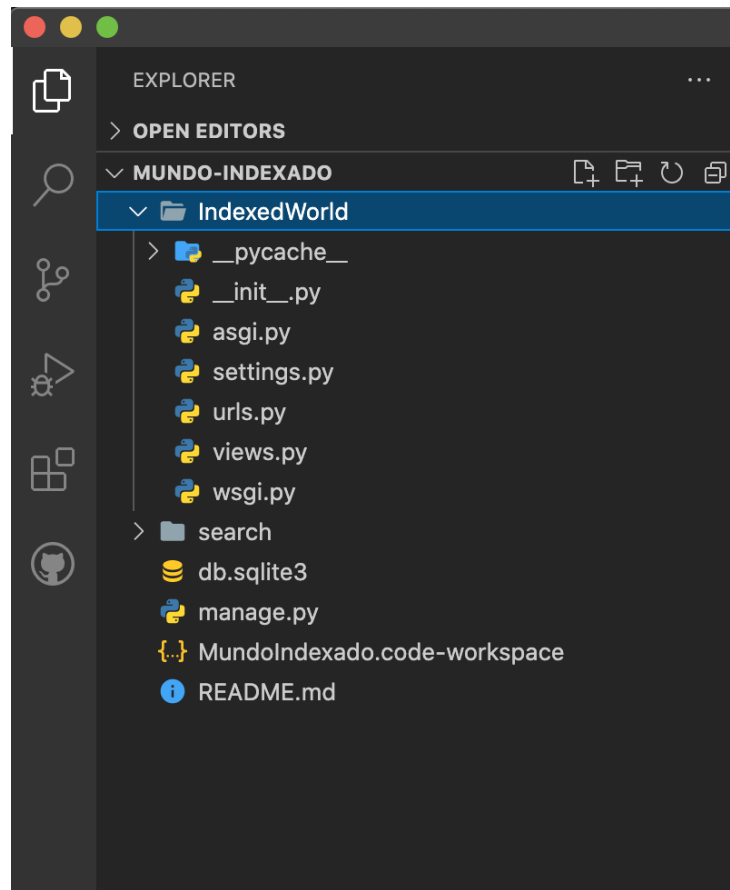
## 6. ANEXO

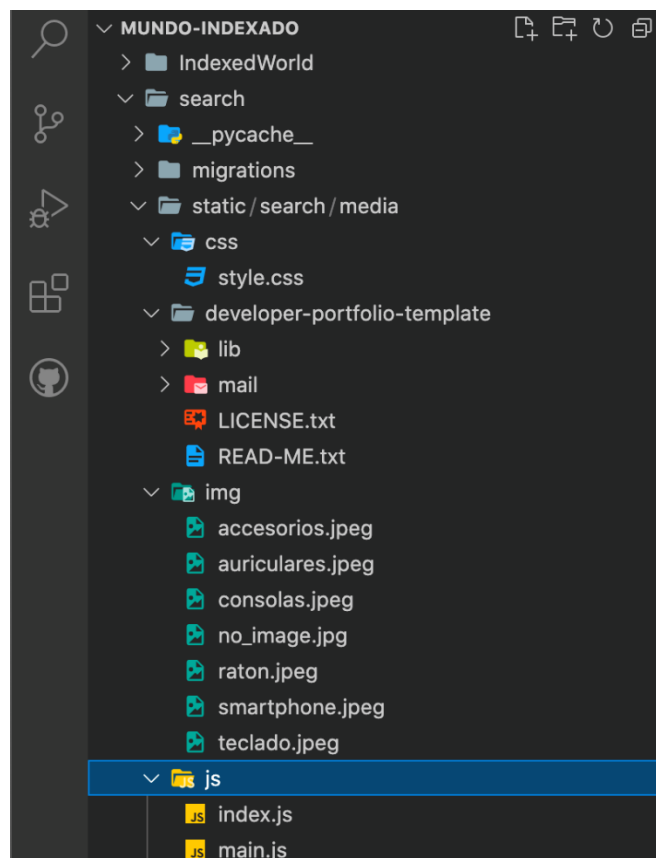
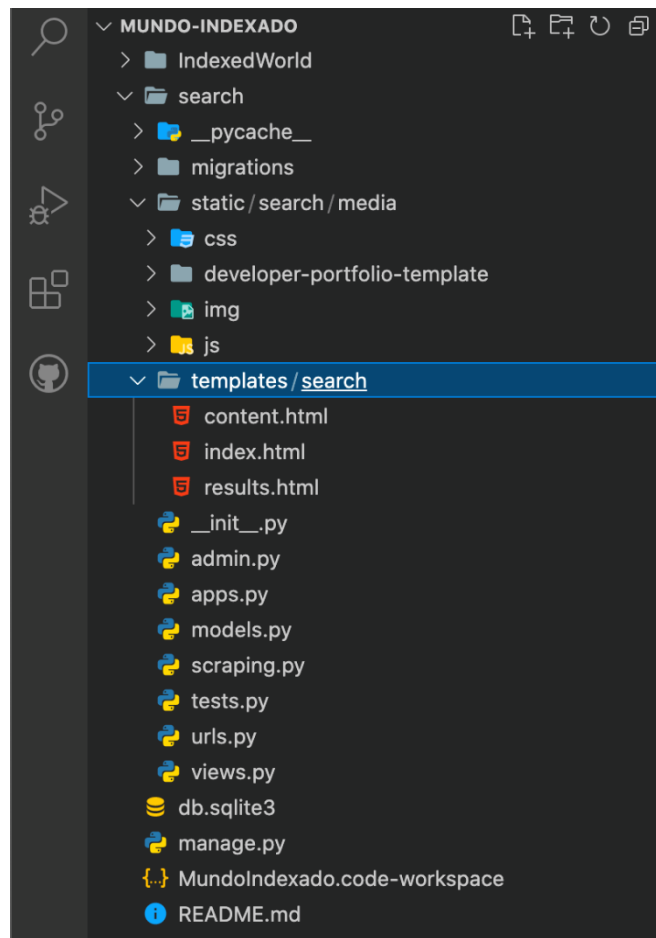
En las siguientes imágenes podemos ver la estructura de directorios y ficheros que componen nuestro proyecto.

Para ver el contenido de esos ficheros o incluso descargarlo todo solo hay que acceder al siguiente enlace que te llevará al repositorio de GitHub dónde tenemos ubicado nuestro proyecto:

<https://github.com/neifors/Mundo-Indexado>







De todos estos ficheros y directorios, hay muchos de ellos reutilizados o se nos han dado hechos por defecto. Al crear un proyecto en Django se crean los archivos de configuración y la estructura base a partir de la cuál hemos trabajado.

## 7. Bibliografía

<https://www.docs.djangoproject.com/en/3.2/>

<https://www.crummy.com/software/BeautifulSoup/bs4/doc>

<https://docs.github.com/es>

<https://docs.python-requests.org/en/master/>

<https://first-web-scraper.readthedocs.io/en/latest/>

<https://docs.python.org/3.8/library/concurrent.futures.html>

<https://stackoverflow.com>

<https://www.free-css.com/free-css-templates>